

NAME

posix_openpt - open a pseudo-terminal device

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

```
#include <fcntl.h>
```

```
#include <stdlib.h>
```

int

```
posix_openpt(int oflag);
```

DESCRIPTION

The **posix_openpt()** function allocates a new pseudo-terminal and establishes a connection with its master device. A slave device shall be created in */dev/pts*. After the pseudo-terminal has been allocated, the slave device should have the proper permissions before it can be used (see **grantpt(3)**). The name of the slave device can be determined by calling **ptsname(3)**.

The file status flags and file access modes of the open file description shall be set according to the value of *oflag*. Values for *oflag* are constructed by a bitwise-inclusive OR of flags from the following list, defined in *<fcntl.h>*:

O_RDWR Open for reading and writing.

O_NOCTTY If set **posix_openpt()** shall not cause the terminal device to become the controlling terminal for the process.

O_CLOEXEC Set the close-on-exec flag for the new file descriptor.

The **posix_openpt()** function shall fail when *oflag* contains other values.

RETURN VALUES

Upon successful completion, the **posix_openpt()** function shall allocate a new pseudo-terminal device and return a non-negative integer representing a file descriptor, which is connected to its master device. Otherwise, -1 shall be returned and **errno** set to indicate the error.

ERRORS

The **posix_openpt()** function shall fail if:

| | |
|----------|---|
| [ENFILE] | The system file table is full. |
| [EINVAL] | The value of <i>oflag</i> is not valid. |
| [EAGAIN] | Out of pseudo-terminal resources. |

SEE ALSO

ptsname(3), pts(4), tty(4)

STANDARDS

The **posix_openpt()** function conforms to IEEE Std 1003.1-2001 ("POSIX.1"). The ability to use **O_CLOEXEC** is an extension to the standard.

HISTORY

The **posix_openpt()** function appeared in FreeBSD 5.0. In FreeBSD 8.0, this function was changed to a system call.

NOTES

The flag **O_NOCTTY** is included for compatibility; in FreeBSD, opening a terminal does not cause it to become a process's controlling terminal.

AUTHORS

Ed Schouten <ed@FreeBSD.org>